GOCUP 2008/2009 – Report

Daniel Maturana (dimatura@puc.cl)

March 16, 2010

1 Methodology

I experimented with various different (meta) heuristics, such as genetic algorithms, estimation of distribution algorthms and variations on greedy procedures. Given enough time, these methods would tend to yield the same set of optima, which I give below.

With these optima in hand, I worked on improving the computational time required to consistently achieve these solutions. It was determined that the best method was simple greedy hill climbing via "vertex substitution" and random restart, i.e. an initial random solution is greedily improved via swaps of cities between the set of cities considered as medians and non-medians until a local minima in cost is reached. The procedure is repeated until 28 consecutive runs fail to improve the best solution or 82 times in total, whichever happens first. These values were chosen to result in the optima given below with approximately .95 probability, but may be specified by the user.

The algorithm was implemented in C++ with special attention to performance. Various of the optimizations suggested by M. Resende y R. Wernicke ('On the implementation of a swap based local search procedure for the p-median problem', Proceedings of the fifth workshop on Algorithm Engineering, 2003) were used.

On an 2GHz Intel Pentium IV with 2GB of RAM 82 independent runs of the algorithm take approximately 6 seconds, though the algorithm often ends earlier. (It can't be predicted, given its stochastic nature). Therefore the complete set of optima can usually be computed in less than 20 seconds.

2 Results

(BONUS: see included png files for a visualization!)

1. Medians: [99, 502, 312, 279, 802, 472, 133, 796, 105, 876] Cost: 6203859
2. Medians: [663, 217, 28, 825, 791, 932, 456, 261, 206, 755] Cost: 6137625
3. Medians: [908, 965, 652, 46, 852, 615, 49, 863, 54, 411] Cost: 6273000
4. Medians: [417, 208, 306, 439, 435, 67, 425, 976, 945, 662] Cost: 6182962
5. Sum 24797446

3 Compilation and usage

The program was developed in C++ using GNU g++ in Ubuntu Linux Hardy Heron. It was also verified to compile and run under Windows XP using the GNU MingW toolchain. It has no external library dependencies. To compile, simply run `make`. The program is named `pmedian` and accepts various arguments. The simplest usage is `pmedian <number>` where number is the problem to solve. It includes various more options which can are documented in the help message which is displayed with `pmedian -h`.